

Database Purger Operations Guide

National Weather Service
Office of Hydrologic Development
August 7, 2003

Table of Contents

1. Overview
2. Selective Purging Based On Station Responsibility
3. Delete of Associated Flat Files
4. Input/Output Summary
5. Outline of Logic

Appendix A. Sample Log File

1.0 Overview

The database purger, known as db_purge, is the application used to purge old records from the Integrated Hydrologic Forecast System (IHFS) database tables. The tables can be classified as follows:

(1) Dynamic Data Tables

(2) TextProduct

(3) VTECevent

The tables which db_purge deletes from are listed in the log file appearing in Appendix A.

For the dynamic data tables, db_purge reads the list of table names in which a purge is to be performed from the PurgeDynData table. The number of hours to retain and the time column name are also read from this table. If the value in the specified time column of the dynamic data table is earlier than the value determined by :

current time - number of hours to retain

then the record is deleted.

The delete from the TextProduct table compares the number of versions of each product found in the TextProduct table against the number of versions to be saved which is read from the num_versions field of the PurgeProduct table. If there are more product versions than specified by the num_versions field, the oldest products are deleted from the TextProduct table until the number of products equals num_versions.

The delete from the VTECevent table begins by determining the latest product time for each geographic event identifier. A geographic event identifier is either a forecast point identifier, a forecast group identifier or a county identifier. All three types of identifiers are supported by the RiverPro application which performs the VTEC (Valid Time Event) formatting. Then for each identifier, records are purged which are older than the value (product time - VTEC_PURGE_TIME) AND not equal to the latest product time. Note that the value of VTEC_PURGE_TIME is currently hard coded to 30 days.

The HydroBase application allows the user to manage the values of the purge parameters for the dynamic data and TextProduct tables through its menu. The proper window can be found by choosing the “Data Ingest” option from the main menu and then choosing “Purge Parameters ...” from the submenu.

The db_purge application is run from the cron. It is normally run once per day. At WFOs it is

normally run at 0745 z. At RFCs, it is normally run at night during hours of light system load.

It should be noted that there is another set of scripts used for purging, `purge_files` and `purge_mpe_files`. These scripts purge flat files from directories under the `/awips/hydroapps` tree and do not use the IHFS DB. Directories containing log files, MPE (Multisensor Precipitation Estimator) output files and Stage1 decoded files are examples of files purged by these scripts.

The following sections focus on the purging of data from the dynamic data tables. The purging of the `TextProduct` and `VTECevent` tables are not discussed further.

2.0 Selective Purging Based on Station Responsibility

WFOs need to store dynamic data in the Informix database for stations in their primary area of responsibility as well as for stations for which they have backup responsibility. Prior to Bld OB2, db_purge made no distinction between these “primary” and “backup” stations. This led to a problem with the dynamic tables containing large numbers of records.

This modification adds the capability to delete records based on retention values for “primary” and “backup” stations. Whether a station is “primary” or “backup” is determined by comparing the hsa field in the Admin table (which defines the primary office) with the hsa field of the Location table record (which defines the office associated with the given station). If the two fields match, then the station is treated as a “primary” station.

This modification allows WFOs to hold more records for their “primary” stations and fewer records for their “backup” stations. The change was implemented in part by a new .Apps_defaults token, so it can be turned off at RFCs and thus not affect their db_purge. The new token is db_purge_backup_retention_use, and it is discussed in more detail in section 4.1.

The PurgeDynData table specifies the purge parameters for the dynamic tables. In Build OB2, it was changed as follows:

- renamed field num_hours to num_hours_host (retention value for primary stations)
- added field num_hours_backup (retention value for backup stations)
 - default value = 72 (if num_hours_host >= 72)
 - = num_hours_host (if num_hours_host < 72)

The HydroBase application was modified for Build OB2 to allow the user to set the new backup retention parameter. The GUI for changing the values is found in the Data Ingest/Purge Parameters GUI under the column heading “Backup Areas”.

A special situation introduced by the new logic is that not all tables being purged by db_purge contain an lid field. Tables such as PerfLog, DPARadar and DPAAadapt do not have the lid field. To get around this problem, a test for an Informix -217 error (column name not found) when preparing the delete_cursor statement has been added. If this return code is generated, a message is printed to the log and the purge uses the host retention value for the table.

The UnkStn and UnkStnValue tables which store data for unknown stations, have an lid field but all records appearing in these tables by definition do not appear in the Location table. For these tables, the delete is done using the backup retention value.

For the ProductLink table, the delete is done using the backup retention value.

3.0 Delete of Associated Flat Files

There are three directories from which db_purge deletes flat files. The delete of the files is tied to the delete of associated records in tables. This assures that the data in the database is consistent with the data stored in flat files. As a record is deleted from the database, the flat file corresponding to that record is deleted. The following is a list of directories (specified by token names) and the related tables for which db_purge deletes flat files:

Token	Default Directory	Table
-----	-----	-----
dpa_grid_dir	/awips/hydroapps/precip_proc/local/data/stage1_decoded	DPARadar
pproc_s2_grid_dir	/awips/hydroapps/precip_proc/local/data/stage2	Stage2Result
ofs_griddb_dir	/awips/hydroapps/rfc/nwsrfs/ofs/files/oper/griddb	Stage3Result

The dpa_grid_dir token “points” to the directory containing the Stage1 decoded files. Each file in this directory is 68652 bytes (131x131x4). The pproc_s2_grid_dir directory contains Stage2 related files. The ofs_griddb_dir directory contains the QPE in xmrg file format output from MPE.

In the case of the DPARadar table, each record contains the actual filename of the file associated with the record. For the other two tables, the filename is created from the value of the obstime field appearing in the record.

4.0 Input/Output Summary

4.1 Input

Input to the db_purge process includes the .Apps_defaults tokens listed in the previous section and the PurgeDynData table. The desired retention periods for primary and backup stations are read from the PurgeDynData table. Each dynamic table having records purged by db_purge appears as a record in this table along with the column name to which the retention period is applied.

The following .Apps_defaults tokens are used by db_purge:

db_purge_log_dir : /awips/hydroapps/whfs/local/data/log/db_purge

Location of db_purge log files.

db_purge_backup_retention_use : ON

This token was introduced in Build OB2.

If on, then db_purge will check the lid of each station to determine if the station is primary or backup. If primary station, then num_hours_host retention will be used to determine which records to retain. If backup station, then num_hours_backup retention will be used.

If off, then db_purge will use the num_hours_host field to determine the number of hours to retain for all stations in all tables and will delete records in the same manner as prior to Bld OB2.

It is expected that WFOs will want the value to be ON while RFCs will want the value OFF.

The user can consult Section 3 of the SHEF Decoder Operations Guide for a full description of the concept and use of .Apps_defaults tokens.

4.2 Output

Each time db_purge is run, a log file is created (see sample in Appendix A) and a record is written to the PerfLog table. Each record in the PerfLog table contains the start_time of the db_purge run, the total number of records deleted from the tables during the run and the elapsed time of the run.

5.0 Outline of Logic

- (1) Open Database
- (2) Read db_purge_backup_retention_use token value
- (3) If (backup_retention_use = ON) then
 read hsa value from Admin table
 If (error reading hsa value) then
 change backup_retention_use to OFF
- (4) Declare and Open purgedyn_cursor for reading records from PurgeDynData table
- (5) Print column headers to log file
- (6) For (each record in the PurgeDynData table)
 - (7) Create “where clause” [WHERE timecol < (current_time - num_hours_XXX)]
 If (backup_retention_use = OFF) then
 use num_hours_host
 Else
 If (num_hours_host > num_hours_backup) then
 use num_hours_backup
 Else
 use num_hours_host
 End If
 (these statements create the least restrictive WHERE clause which will allow for further testing before deleting a record)
 - (8) Delete files associated with records from Stage2Result, Stage3Result, DPARadar tables
 - (9) Create delete cursor statement [SELECT lid,timecol from tablename WHERE ...]
 - (10) Prepare delete cursor
 If (SQL return code = -217 i.e. lid field not part of record) then
 create and prepare delete cursor again without lid field
 - (11) Declare and Open delete cursor
 - (12) Create delete statement [DELETE FROM tablename WHERE CURRENT OF cursorname]
 - (13) Prepare delete statement

```

(14) Set delete_flag = 1

(15) For each record fetched from table (store lid, timeval OR timeval only)
    (16) If (SQL return code = 0) then

        (17) If (backup_retention_use = ON AND record has lid) then

            (18) If (previous lid different than current lid) then

                Set delete_flag = 0

            (19) Get hsa value for station from Location table

            (20) If (location.hsa = admin.hsa) then

                (21) If (numhr_host > numhr_backup) then
                    (22) If (timeval < delete_time_host) then
                        Set delete_flag = 1
                    Else
                        Set delete_flag = 1
                    End If
                Else
                    (23) If (numhr_host < numhr_backup) then
                        (24) If (timeval < delete_time_backup) then
                            Set delete_flag = 1
                        Else
                            Set delete_flag = 1
                        End If
                    End if
                End if

            End if

        End if

    End If

    (25) If(delete_flag = 1) then
        delete record
    Else
        If (SQL return code = 100) then

            (26) Commit Work and break out of loop

```



```

Else
    (27) Print error message
    (28) Rollback and break out of loop
End If
End If
End For
(29) Print number of records deleted, etc. to log file
(30) If (table does not have lid field) then print message to log
End For (PurgeDynData table records)
(31) Close and Free cursors
(32) Delete products from TextProduct table
(33) Delete records from the VTECevent table
(34) Write record to PerfLog table
```

Appendix A: Sample Log File

Log files are created for each run with the names db_purge.log.MMDD_HHMM in the directory defined by the db_purge_log_dir token. Normally, this token is set to /awips/hydroapps/whfs/local/data/log/db_purge. Note that the time spent in deleting records from each table can be determined by subtracting minutes/seconds from consecutive lines of the time cutoff column(s). The time cutoff is derived from the number of hours to retain as specified in the PurgeDynData table. The number of hours to retain is defined in whole hours, so this value is compared to the system clock to determine the time cutoff. To ascertain the amount of time spent on each table, one can ignore the hours element of the time cutoff. For example, it can be seen below that db_purge spent :03 seconds deleting 100 records from the agricultural table (45:07 - 45:04).

IHFS Database Purge AWIPS ROB2 January 14,2003

begin time = Jan 17, 2003 - 07:45:03

database name = hd_ob2ounx

Using num_hours_backup field to determine retention for backup stations

hsa (from Admin table) = OUN

number deleted	table name	time column	time cutoff (host)	time cutoff (backup)
100	agricultural	obstime	2003-01-07 01:45:04	2003-01-13 01:45:04
262	alertalarmval	postingtime	2003-01-16 13:45:07	2003-01-16 13:45:07
0	commentvalue	validtime	2003-01-02 01:45:08	2003-01-14 01:45:08
531	contingencyvalue	validtime	2003-01-02 01:45:09	2003-01-14 01:45:09
3003	curprecip	obstime	2003-01-13 13:45:13	2003-01-14 01:45:13
0	discharge	obstime	2003-01-15 14:45:50	2003-01-15 12:45:50
441	dpaadapt	obstime	2003-01-15 13:45:50	2003-01-15 13:45:50
	-- table does not have lid field - host cutoff value used			
441	dparadar	obstime	2003-01-15 13:45:53	2003-01-15 13:45:53
	-- table does not have lid field - host cutoff value used			
0	evaporation	obstime	2003-01-02 01:45:55	2003-01-14 01:45:55
0	fcstdischarge	validtime	2003-01-02 01:45:55	2003-01-14 01:45:55
480	fcstheight	validtime	2003-01-02 01:45:55	2003-01-14 01:45:55
0	fcstother	validtime	2002-12-18 01:46:07	2003-01-14 01:46:07
46	fcstprecip	validtime	2003-01-02 01:46:07	2003-01-14 01:46:07
0	fcsttemperature	validtime	2003-01-02 01:46:09	2003-01-14 01:46:09
0	fishcount	obstime	2003-01-02 01:46:09	2003-01-14 01:46:09
28	floodts	obstime	2001-10-24 01:46:09	2003-01-14 01:46:09
0	fpprevprod	producttime	2002-12-18 01:46:33	2003-01-14 01:46:33
0	gatedam	obstime	2003-01-02 01:46:33	2003-01-14 01:46:33
0	ground	obstime	2003-01-02 01:46:33	2003-01-14 01:46:33
1251	height	obstime	2003-01-02 01:46:33	2003-01-14 01:46:33
0	ice	obstime	2003-01-02 01:49:54	2003-01-14 01:49:54
0	lake	obstime	2003-01-02 01:49:54	2003-01-14 01:49:54
0	moisture	obstime	2003-01-02 01:49:54	2003-01-14 01:49:54
0	pairedvalue	validtime	2003-01-02 01:49:54	2003-01-14 01:49:54
276	perflog	start_time	2003-01-12 01:49:54	2003-01-14 01:49:54
	-- table does not have lid field - host cutoff value used			
0	power	obstime	2003-01-02 01:49:55	2003-01-14 01:49:55
2039	precip	obstime	2003-01-02 01:49:55	2003-01-14 01:49:55
0	pressure	obstime	2003-01-02 01:55:21	2003-01-14 01:55:21
3723	procprecip	obstime	2003-01-14 01:55:21	2003-01-14 01:55:21
0	procvalue	obstime	2003-01-02 01:55:46	2003-01-14 01:55:46
5640	productlink	producttime	2003-01-07 01:55:46	2003-01-14 01:55:46
0	pseudogageradarval	obstime	2003-01-15 13:57:56	2003-01-15 13:57:56

```

-- table does not have lid field - host cutoff value used
2568 radiation          obstime      2003-01-02 01:57:56  2003-01-14 01:57:56
  31 rejecteddata      validtime    2003-01-14 01:59:42  2003-01-14 01:59:42
    0 riverstatus      validtime    2003-01-12 01:59:42  2003-01-14 01:59:42
2160 rwbiasdyn          obstime      2003-01-13 01:59:42  2003-01-14 01:59:42
-- table does not have lid field - host cutoff value used
  216 rwradarresult     obstime      2003-01-13 01:59:50  2003-01-14 01:59:50
-- table does not have lid field - host cutoff value used
    24 rwresult         obstime      2003-01-13 01:59:51  2003-01-14 01:59:51
-- table does not have lid field - host cutoff value used
    0 rwverif          obstime      2003-01-13 01:59:51  2003-01-14 01:59:51
    0 s2gageradarval    obstime      2003-01-16 01:59:51  2003-01-16 01:59:51
    0 s3gridmanip       obstime      2003-01-07 01:59:51  2003-01-14 01:59:51
-- table does not have lid field - host cutoff value used
  101 snow              obstime      2003-01-02 01:59:51  2003-01-14 01:59:51
    0 stage2result      obstime      2003-01-12 01:59:55  2003-01-14 01:59:55
-- table does not have lid field - host cutoff value used
    0 stage3result      obstime      2003-01-12 01:59:55  2003-01-14 01:59:55
-- table does not have lid field - host cutoff value used
6788 temperature       obstime      2003-01-02 01:59:55  2003-01-14 01:59:55
   42 unkstn            postingtime  2002-07-21 02:03:51  2003-01-14 02:03:51
    0 unkstnvalue       obstime      2003-01-15 14:03:52  2003-01-15 14:03:52
    0 waterquality      obstime      2003-01-02 02:03:52  2003-01-14 02:03:52
    0 weather           obstime      2003-01-02 02:03:52  2003-01-14 02:03:52
2140 wind              obstime      2003-01-13 02:03:52  2003-01-14 02:03:52
    1 yunique           obstime      2003-01-02 02:04:14  2003-01-14 02:04:14
-----

```

32332 records deleted from 51 tables

Begin PurgeProduct:

```

KTSARR3TUL - None deleted; 0 products currently <= 0 max.
KTUARR6TUR - None deleted; 0 products currently <= 0 max.
KLUBRR4LBB - None deleted; 0 products currently <= 0 max.
KTUARVRTUR - None deleted; 0 products currently <= 0 max.
KTUARVFOKC - None deleted; 0 products currently <= 0 max.
KTUARVFSHV - None deleted; 0 products currently <= 0 max.
.
.
.
KLUBRR1LBB - None deleted; 0 products currently <= 0 max.
KWOHRRSOUN - None deleted; 0 products currently <= 0 max.
KFWDRR3ACT - None deleted; 0 products currently <= 0 max.
KFWRRR9FWR - None deleted; 0 products currently <= 0 max.

```

Begin Purge VTECevent table

```

BLAINE,OK - None deleted;
BRYAN,OK - None deleted;
CLAY,TX - None deleted;
KINGFISHER,OK - None deleted;
LOGAN,OK - None deleted;
WICHITA,TX - None deleted;
WILBARGER,TX - None deleted;
WOODS,OK - None deleted;
End Purge VTECevent table

```

```

end time = Jan 17, 2003 - 08:04:18
Elapsed Time = 1155 sec
              = 19.25 min

```